sequenceAnalysis.py User Guide

2015 UCSC iGEM Team

September 18, 2015

Table of Contents

List of Figures

General Overview

1.1 What is it for?

The sequenceAnalysis program is a tool for providing efficient, large scale analysis of amino acid sequences. It works very much like the software ProtParam designed by ExPASy. However, unlike ProtParam, this program has the advantage of reading and computing the physical and chemical properties of multiple proteins all at once.

1.2 Program Specifications

This program was written using the most recent version of the Python programming language, Python 3.4.3. It is designed to work as module, which can be imported to any program to access all or specific functions that the user requires. An added benefit of the modular design is that it can be easily edited to provide further functionality. At this point, this program is made up of two classes: ProteinParam and FastAReader.

Class ProteinParam

This class was written by UCSC iGEM team members Cristian Camacho, Jairo Navarro, and Raymond Bryan. It was developed in the course BME 160: Research Programming in the Life Sciences, and serves as the backbone for two other programs that are being submitted: CodonBiasGenerator and FOCUS.

2.1 Attributes

- aa2mw : A dictionary of the molecular weights of all 20 amino acids
- mwH20 : A float value corresponding to the molecular weight of water
- aa2abs280 : Dictionary of the absorbance values of Tyrosine, Tryptophan and Cysteine at a wavelength of 280 nm.
- aa2chargePos : Dictionary of the positive charge values of Lysine, Arginine and Histidine
- aa2chargeNeg : Dictionary of the negative charge value of Aspartic Acid, Glutamic Acid, Cysteine and Tyrosine.
- aaNterm : Float value of the charge
- aaCterm : Float value of
- validAA : An empty dictionary which will contain the counts of valid amino acids in a specific protein sequence.

2.2 Methods

- aaCount( ) : Iterates through the amino acid sequence and returns a single integer count of valid amino acid characters found.
- aaComposition ( ) : Returns the validAA dictionary with the valid amino acids and their counts for a specific protein.
- pI ( ) : Estimates the theoretical isoelectric point of a protein by iterating through every pH value until it finds the one that results in a net charge that is closest to zero.

- charge ( ) : Calculate the net charge at a particular pH, using the pKa of each charged Amino acid and the Nterminus and Cterminus
- molarExtinction ( ) : Estimates the molar extinction coefficient based on the number and extinction coefficients of tyrosines, tryptophans, and cysteines.
- massExtinction ( ) : Calculates the mass extinction by dividing the molar extinction value by the molecular weight of the corresponding protein.
- molecularWeight ( ) : Calculates a proteins molecular weight by summing the weights of the individual Amino acids and excluding the waters that are released with peptide bond formation.

Class FastAreader

This program was developed by Professor David Bernick of UC Santa Cruz, for the upper- division course BME 160: Research Programming in the Life Sciences. This class is what allows the sequenceAnalysis module to read and calculate the characteristics of multiple protein sequences at the same time, as long as they are in the FASTA format.

3.1 Attributes

- fname : The initial file name to be ready by FastAreader.

3.2 Methods

- doOpen ( ) : Checks if a file name is given to FastAreader, and if not, waits for a file to be given through system.in. This function provides command line usability.
- readFasta ( ) : Using filename given in init, returns each included FastA record as 2 strings - header and sequence. If a filename is not provided, std.in is used. Whitespace is removed, no adjustment is made to sequence contents. The initial '>' is removed from the header.

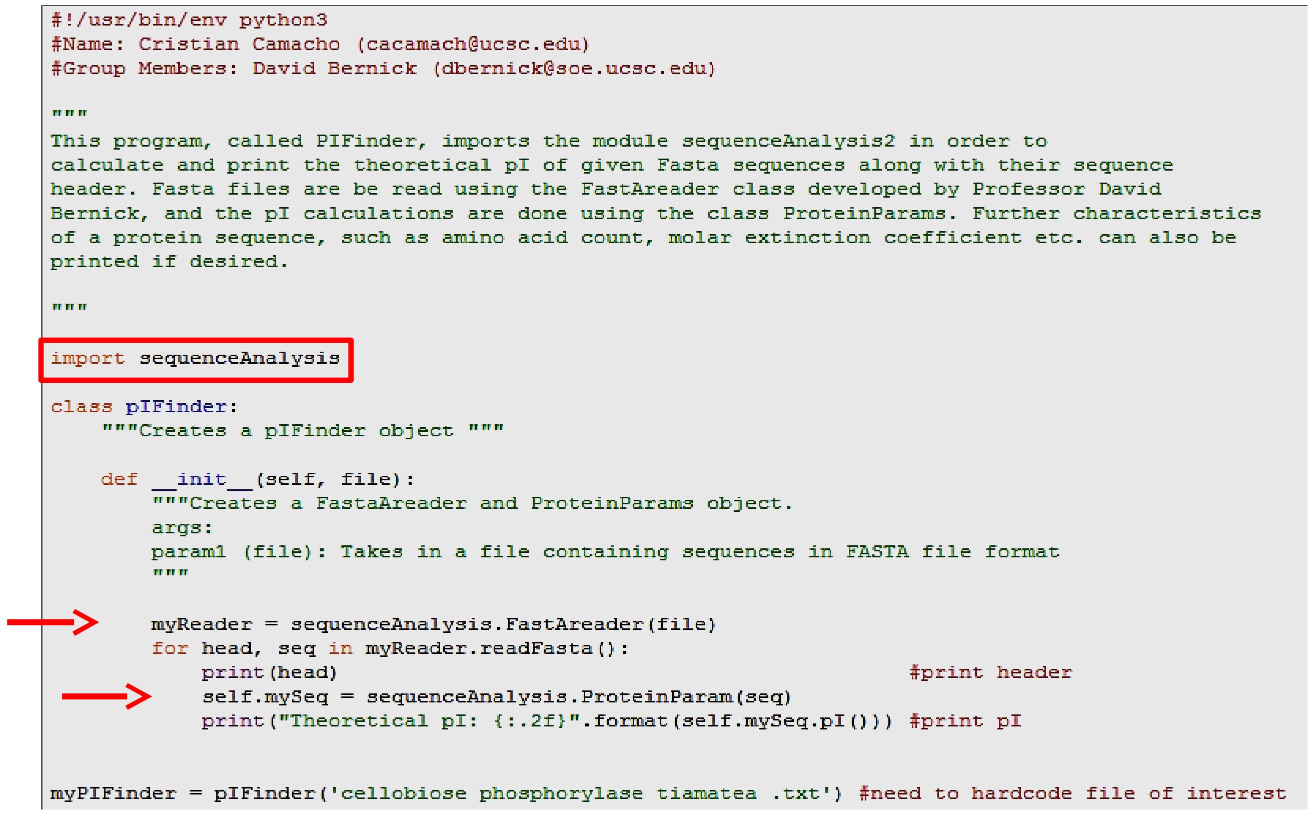
Importing the Module for Use

The use of the sequenceAnalysis module is fairly simple. Here are the following steps for making use of it in your programs:

1. Download the file named sequenceAnalysis from either the UCSC iGEM 2015 wiki, or the 2015 iGEM GitHub.
2. Important, save the file in the same directory as the script that you are writing.
3. Make sure to include the following line "import sequenceAnalysis" before writing any code for your new program.
4. In order to use a function from either the ProteinParam class or FastAreader class, you must create an object for that class.
5. Then you can use that object to access any of the available function from that class.

Below is an example program which illustrates the above steps and explains how to access one or more of the functions from ProteinParam.

4.1 Ex) pIFinder.py



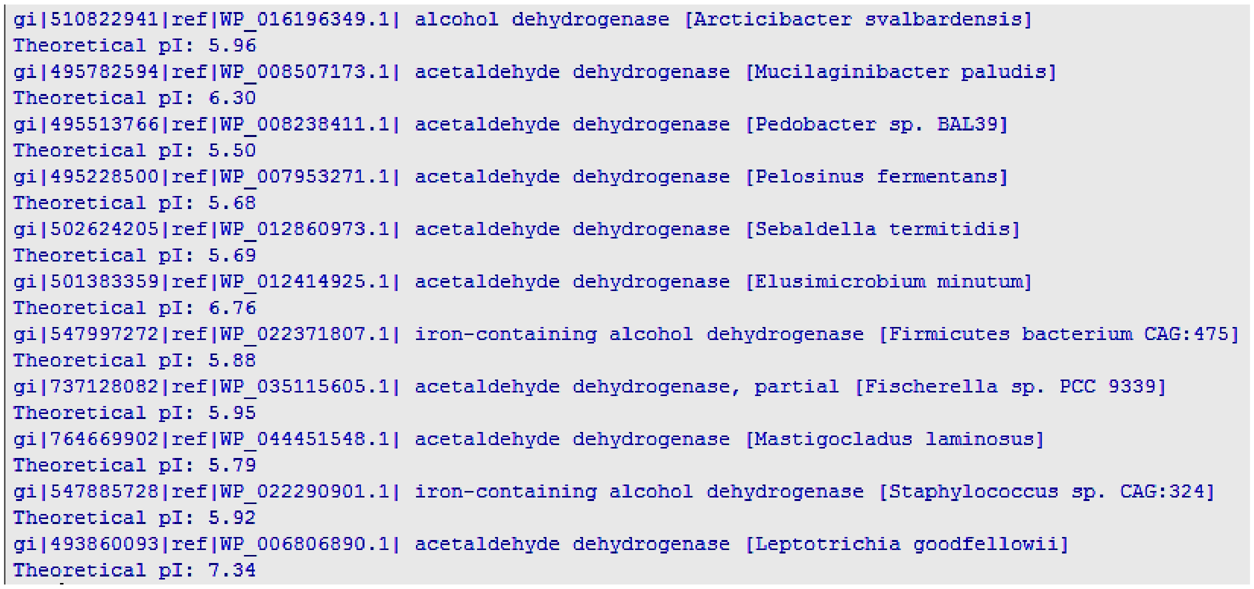
```
#!/usr/bin/env python3
#Name: Cristian Camacho (cacamach@ucsc.edu)
#Group Members: David Bernick (dbernick@soe.ucsc.edu)

"""
This program, called PIFinder, imports the module sequenceAnalysis2 in order to
calculate and print the theoretical pI of given Fasta sequences along with their sequence
header. Fasta files are be read using the FastAreader class developed by Professor David
Bernick, and the pI calculations are done using the class ProteinParams. Further characteristics
of a protein sequence, such as amino acid count, molar extinction coefficient etc. can also be
printed if desired.

"""

import sequenceAnalysis

class pIFinder:
    """Creates a pIFinder object """

    def __init__(self, file):
        """Creates a FastaAreader and ProteinParams object.
        args:
        param1 (file): Takes in a file containing sequences in FASTA file format
        """

        myReader = sequenceAnalysis.FastAreader(file)
        for head, seq in myReader.readFasta():
            print(head)                                  #print header
            self.mySeq = sequenceAnalysis.ProteinParam(seq)
            print("Theoretical pI: {:.2f}".format(self.mySeq.pI())) #print pI


myPIFinder = pIFinder('cellobiose phosphorylase tiamatea .txt') #need to hardcode file of interest
```

Figure 1: This program is known as pIFinder, which specifically utilizes the pI ( ) method of the class ProteinParam and the FastAreader class to calculate and print the isoelectric point of given protein sequences with their respective headers.

Notice that in the red box there is the line "import sequenceAnalysis", which signifies that the capabilities of sequenceAnalysis are now available to your program. Also, notice that the lines that which the two arrows are pointing to are responsible for creating a FastAreader object and a ProteinParam object.

Test Files and Results

In order to test whether your program is working correctly, we have provided a test file that includes eleven FASTA formatted protein sequences. This test file can be found on the 2015 UCSC iGEM wiki under the name “sequenceAnalysisTest.txt”. Make sure that this file is also saved in the same directory as sequenceAnalysis and the program that you are writing. Furthermore, also make sure that whether you are hardcoding the file name into your code, or submitting it via the command line, that the file name matches exactly the way it is written.



```
gi|510822941|ref|WP_016196349.1| alcohol dehydrogenase [Arcticibacter svalbardensis]
Theoretical pI: 5.96
gi|495782594|ref|WP_008507173.1| acetaldehyde dehydrogenase [Mucilaginibacter paludis]
Theoretical pI: 6.30
gi|495513766|ref|WP_008238411.1| acetaldehyde dehydrogenase [Pedobacter sp. BAL39]
Theoretical pI: 5.50
gi|495228500|ref|WP_007953271.1| acetaldehyde dehydrogenase [Pelosinus fermentans]
Theoretical pI: 5.68
gi|502624205|ref|WP_012860973.1| acetaldehyde dehydrogenase [Sebaldella termitidis]
Theoretical pI: 5.69
gi|501383359|ref|WP_012414925.1| acetaldehyde dehydrogenase [Elusimicrobium minutum]
Theoretical pI: 6.76
gi|547997272|ref|WP_022371807.1| iron-containing alcohol dehydrogenase [Firmicutes bacterium CAG:475]
Theoretical pI: 5.88
gi|737128082|ref|WP_035115605.1| acetaldehyde dehydrogenase, partial [Fischerella sp. PCC 9339]
Theoretical pI: 5.95
gi|764669902|ref|WP_044451548.1| acetaldehyde dehydrogenase [Mastigocladus laminosus]
Theoretical pI: 5.79
gi|547885728|ref|WP_022290901.1| iron-containing alcohol dehydrogenase [Staphylococcus sp. CAG:324]
Theoretical pI: 5.92
gi|493860093|ref|WP_006806890.1| acetaldehyde dehydrogenase [Leptotrichia goodfellowii]
Theoretical pI: 7.34
```

Figure 2: The results for the test file.